Syntactic Characterizations of Belief Change Operators

Alvaro del Val
Robotics Lab
Computer Science Department
Stanford University
Stanford, CA 94305
e-mail: delval@cs.stanford.edu

Abstract

We provide syntactic characterizations for a number of propositional model-based belief revision and update operators proposed in the literature, as well as algorithms based on these characterizations.

1 Introduction

In this paper, we provide syntactic characterizations and algorithms for a number of belief change operators proposed in the literature. We already characterized Winslett's 'possible models approach' (PMA) update operators in [del Val, 1992b], where we explored in depth some of the operators in the PMA family, provided algorithms to compute them and experimentally showed that they could be of practical value for (small) updates of quite large databases. In this paper, we show how other operators can be characterized in a very similar way, and show how to design algorithms for computing the result of applying these operators to disjunctive, negation and conjunctive normal form (DNF, NNF and CNF, respectively) databases, which return a database in the same format.

The interest of these syntactic characterizations goes beyond, we believe, the usefulness of the algorithms that can be immediately derived from them. All the operators we discuss are based on some notion of minimal change, and the similarities among the various characterizations we provide suggests that our techniques can be easily extended to other belief change operators based on this notion that might be proposed in the future. as well as to variants of the operators discussed here (e.g. prioritized versions, as discussed in [del Val. 1992b; del Val, 1993]). Because of their formal nature and relative simplicity, the characterizations could also prove useful for the design of improved algorithms, facilitate proofs of their correctness, help identify syntactic restrictions with a positive impact on complexity, and help define useful notions of "approximate belief change" (see also [del Val, 1992a] on this last point).

Much AI work on belief change has taken as starting point the "AGM approach" to belief revision proposed by Alchourron, Gardenfors and Makinson [Alchourron et al., 1985; Gardenfors, 1988]. Katsuno and Mendelzon [1991a] have recently suggested, however, that at least two different belief change operations should be distinguished, revision and update. Loosely speaking, the former says that the beliefs may have been wrong and in need of revision, whereas the latter says that the beliefs were correct, but the world has in the meanwhile evolved and the beliefs must be updated. Both types of belief change have been characterized by relation to some sets of properties or "postulates" (the AGM and the KM postulates, respectively). As shown by Katsuno and Mendelzon [1991b; 1991a], most AGM-like revision operators are based on a "global" order of preferences associated to each database, while all (KM) update operators are based on a "pointwise" order in which each model of the database has an associated ordering. As we will see, this distinction is very directly reflected in our characterizations. In fact, except in the case of DNF databases, it has a direct impact on the complexity of the algorithms we present, an impact which is directly related to the fact that in revision some models of the original database are pruned and fail to "generate" any models of the revised database. This added complexity can however be substantially reduced for Horn databases.

The structure of this paper is as follows. The first three sections of the paper are devoted to update operators, by which we mean operators satisfying the KM postulates; specifically, we consider Winslett's [1988] set inclusion based operator and Forbus' [1989] cardinality based operator The following sections discuss revision operators, by which we mean operators satisfying at least the "basic" AGM postulates for revision (that is, (RI)-(K4) in the notation of [Katsuno and Mendelzon, 1991a]; in particular, they all have the property that ψ revised with μ equals $\psi \wedge \mu$ whenever this formula is satisfiable), specifically, we consider Dalal's [1988a] cardinality based approach and Satoh's [1988] set inclusion based proposal, as well as the proposals of [Weber, 1986] and [Borgida, 1985] Related work is discussed in the concluding section.

In the rest of the paper, we assume a propositional language with a finite set V of symbols. Update operators are represented by $\diamond$, and revision operators with o, both possibly subscripted. ψ always denotes the database and μ the update formula, both of which are assumed to be satisfiable. If ψ is in CNF or NNF, it consists of clauses or top level conjuncts $c_i, c_j, \ldots$. $\mathrm{DNF}(\psi)$ represents some formula in disjunctive normal form equivalent to ψ and consisting of the conjunctions of literals $\psi_i, \psi_j, \ldots$ We require that all these conjunctions be satisfiable. μ is assumed to be in DNF, and its disjuncts will be denoted by $\mu_i, \mu_j, \ldots$ Literals are represented by $l, l_i, l_j, \ldots$. With a slight abuse of notation, if there is no ambiguity a conjunction of literals will be treated interchangeably as a set of literals, and a DNF (CNF, NNF) formula as a set of disjuncts (conjuncts). We also use, for any formula or set of formulas ϕ, $Prop(\phi)$ for the set of propositional symbols occurring in ϕ, and $Mod(\phi)$ for the models of ϕ. Finally, for any set S and binary relation $\leq$, we use $Min(S, \leq)$ for the set of elements of S minimal under $\leq$.

The following definitions will also be useful. Define the "difference" between two models I and J as $Diff(I,J) = \{p \in \mathcal{P} \mid I \models p \text{ iff } J \models \neg p\}$, the set of letters on which they differ, and the "distance" between two models as the cardinality of their difference, *i.e.* $Dist(I,M) = |Diff(I,M)|$. Similarly, define the difference between two conjunctions of literals ψ_i and μ_j as the set of literals in ψ_i whose negation is in μ_j, *i.e.* $Diff(\psi_i, \mu_j) = \{l \in \psi_i \mid \neg l \in \mu_j\}$, and define their distance analogously as $Dist(\psi_i, \mu_j) = |Diff(\psi_i, \mu_j)|$

2 Winslett's set inclusion based update

The "possible models approach" to update was proposed in [Winslett, 1988]. As other operators we will discuss, the PMA update operator $\circ_W$ selects a subset of "preferred" models of the update formula as models of the updated database. Specifically, $\circ_W$ collects, for each model M of the original database ψ, the models of the update formula μ which differ in fewest (in the set-inclusion sense) propositional letters from M. Formally, we associate to each model M an ordering $\leq_M^W$ over interpretations defined by: $I \leq_M^W J$ iff $Diff(I,M) \subseteq Diff(J,M)$. The update operator $\circ_W$ is then defined by

$$Mod(\psi \circ_W \mu) = \bigcup_{M \in Mod(\psi)} Min(Mod(\mu), \leq_M^W)$$

Example 1 Let $\psi = (b \wedge c) \vee (\neg a \wedge \neg b \wedge c)$ and $\mu = (a \wedge \neg b) \vee (\neg b \wedge \neg c)$. There are three models of ψ, namely, $M_1 = \{a, b, c\}$, $M_2 = \{\neg a, b, c\}$ and $M_3 = \{\neg a, \neg b, c\}$, and three models of μ, $N_1 = \{a, \neg b, c\}$, $N_2 = \{\neg a, \neg b, \neg c\}$ and $N_3 = \{a, \neg b, \neg c\}$. Only N_1 is minimal with respect to M_1; with respect to M_2 both N_1 and N_2 are minimal, and similarly with respect to M_3. Thus, $Mod(\psi \circ \mu) = \{N_1, N_2\}$. □

As shown in [del Val, 1992b], the PMA update operator can be syntactically characterized as follows. We first need several definitions, motivated below. For $\mu_j, \mu_k \in \mu$, $\psi_i \in \text{DNF}(\psi)$, let

$$\begin{aligned}
\mu^*_{\psi_i}(\mu_j) &= \{\mu_k \in \mu \mid Diff(\mu_j, \psi_i) \not\subseteq \mu_k\} \\
neg_{\psi_i}(\mu_k, \mu_j) &= \{\neg l \mid l \in \mu_k - (\psi_i \cup \mu_j)\} \\
patch_{\psi_i}(\mu_j) &= \bigwedge_{\mu_k \in \mu^*_{\psi_i}(\mu_j)} (\bigvee neg_{\psi_i}(\mu_k, \mu_j))).
\end{aligned}$$

where, by convention, $\bigvee \emptyset = false$, $\bigwedge \emptyset = true$. (I'll drop the subscript ψ_i from now on when using any of these and similar functions, since the context will always make clear the appropriate subscript.)

Theorem 1 [1]

$$\psi \circ_W \mu \equiv \bigvee_{\substack{\mu_j \in \mu \\ \psi_i \in \text{DNF}(\psi)}} (\bigwedge((\psi_i - Diff(\psi_i, \mu_j)) \cup \mu_j) \wedge patch(\mu_j)).$$

The basic idea behind this theorem is as follows. First, we can see each $\psi_i \in \text{DNF}(\psi)$ as a partial model of the database ψ, and update each partial model independently. For each $\psi_i \in \text{DNF}(\psi)$, then, we need to select models of μ which are closest to some model of ψ_i. This is done in two steps. We first select, for each $\mu_j \in \text{DNF}(\mu) = \mu$, the models of μ_j which satisfy as many literals as possible from ψ_i, *i.e.* the set of models of the formula $\delta_{ij} = \bigwedge((\psi_i - Diff(\psi_i, \mu_j)) \cup \mu_j)$. For every $N \in Mod(\delta_{ij})$, there exists $M_N \in Mod(\psi_i)$ such that $Diff(N, M) = Prop(Diff(\psi_i, \mu_j))$. Clearly, no other model of μ_j can be closer to M_N than N, and similarly for models of any $\mu_k \notin \mu^*(\mu_j)$; but this might not be true if we also consider models of some $\mu_k \in \mu^*(\mu_j)$. In the second step, then, we further filter out the selected models of μ_j (which at this point are the models of δ_{ij}), by ensuring that any selected model N is such that the corresponding $M_N \in Mod(\psi_i)$ differs from any model of some $\mu_k \in \mu^*(\mu_j)$ in some letter not in $Prop(Diff(\psi_i, \mu_j))$. This is done by "patching up" δ_{ij} by adding to it the negation of a literal $l_k \in \mu_k$ for each $\mu_k \in \mu^*(\mu_j)$. Clearly, l_k should not be in $Diff(\psi_i, \mu_j)$; and l_k should not be in δ_{ij}, since otherwise $\neg l_k \wedge \delta_{ij}$ would be inconsistent. In other words, $\neg l_k$ must be in $neg(\mu_k, \mu_j)$. If $N \in Mod(\delta_{ij} \wedge \neg l_k)$, then no model of μ_k can be set-inclusion closer to M_N than N. Finally, since we have to choose some l_k for each $\mu_k \in \mu^*(\mu_j)$, there might be many ways to consistently combine these choices, and all possible combinations must be considered. The models (if any) of $patch(\mu_j)$ are exactly the models of some such consistent combination.

3 Forbus' cardinality based update

Forbus [1989] proposes a cardinality-based update operator, which differs from the PMA only in the replacement of PMA's set-inclusion minimization of changes by the minimization of the *number* of model differences. Again, we assign to each interpretation M an ordering $\leq_M^F$ over interpretations according to which $I \leq_M^F J$ iff $Dist(I,M) \leq Dist(J,M)$. Forbus' operator $\circ_F$ is defined by:

$$Mod(\psi \circ_F \mu) = \bigcup_{M \in Mod(\psi)} Min(Mod(\mu), \leq_M^F).$$

We now syntactically characterize the operator $\circ_F$. The reader should notice that the only differences with

[1] This theorem is simply a more compact expression of theorem 1 in [del Val, 1992b]. There, we used $neg(\mu^*(\mu_j))$ for what here would be $\text{DNF}(patch(\mu_j))$. Considering the three cases given in that paper, we have: if $\mu^*(\mu_j) = \emptyset$ then $patch(\mu_j) = true$; if $\text{DNF}(patch(\mu_j)) = \emptyset$, then $patch(\mu_j) \equiv false$; the remaining case can be obtained by using the tautology $[\bigvee_{\theta \in \text{DNF}(\varphi)} (\gamma \wedge \theta)] \equiv \gamma \wedge \varphi$.

respect to the characterization of the PMA operator lie in the selection by $\mu^*_F(\mu_j)$ of the disjuncts in μ to be used by $neg_F(\mu^*_F(\mu_j))$, and in the selection of literals from each $\mu_k \in \mu^*_F(\mu_j)$ to be negated and added to the formula δ_{ij}.

Formally, for $\mu_j, \mu_k \in \mu$, $\psi_i \in \mathrm{DNF}(\psi)$, define:

$$\begin{aligned}
\mu^*_F(\mu_j) &= \{\mu_k \in \mu \mid Dist(\psi_i,\mu_k) < Dist(\psi_i,\mu_j)\} \\
neg_F(\mu_k,\mu_j) &= \{d \mid \exists S \subseteq (\mu_k - \mu_j) : d = \textstyle\bigwedge_{l \in S} \neg l, \\
&\quad\; Prop(S) \cap Prop(\psi_i) = \emptyset, \text{ and} \\
&\quad\; |S| = Dist(\psi_i,\mu_j) - Dist(\psi_i,\mu_k)\} \\
patch_F(\mu_j) &= \bigwedge_{\mu_k \in \mu^*_F(\mu_j)} (\bigvee neg_F(\mu_k,\mu_j)))
\end{aligned}$$

Theorem 2

$$\psi \circ_F \mu \equiv \bigvee_{\substack{\mu_j \in \mu \\ \psi_i \in \mathrm{DNF}(\psi)}} (\bigwedge((\psi_i - Diff(\psi_i,\mu_j)) \cup \mu_j) \wedge patch_F(\mu_j)).$$

4 Update algorithms

The previous results make it trivial to design very efficient algorithms for computing $\circ_W$ and $\circ_F$ for databases represented as a DNF formula or under the 'model checking approach' [Halpern and Vardi, 1991; Grahne and Mendelzon, 1991], in which the database is represented as a set of models. But storing databases in this way will often be unfeasible, so we need methods which will work with more common formats such as CNF and NNF. According to the next theorem, for CNF and NNF databases it suffices to update the subset of the database sharing symbols with the update formula.

Theorem 3 *Let $\circ \in \{\circ_W, \circ_F\}$. Let ψ be an NNF or CNF database, let $\psi_S = \{c_i \in \psi \mid Prop(c_i) \cap Prop(\mu) \neq \emptyset\}$ be the set of clauses or top level conjuncts sharing some propositional symbols with μ, and let $\psi_U = \psi - \psi_S$. Then $\psi \circ \mu \equiv (\psi_S \circ \mu) \wedge \psi_U$.*

This theorem has a dramatic effect on the cost of computing the update. Update formulas will typically be rather short; assuming that any particular symbol occurs only in a few number of clauses or top level conjuncts, this makes the cost of the update largely independent of the total size of the database. In [del Val, 1992b] we presented an algorithm for PMA update based on this theorem, whose worst case complexity (for CNF input and NNF output, and ignoring retrieval costs) is bounded by $O((\prod_{\psi_S} |c_i|)(|\mu|(\mu_{max})^{|\mu|-1}))$ for $|\mu| > 1$.[2] Here $|c_i|$ represents the size (number of literals) of the clause c_i, with the product taken over all clauses in ψ_S; μ_{max} is the maximum size of a disjunct in μ, and $|\mu|$ the number of disjuncts in μ. Since μ will typically be quite small, the crucial factor is clearly $\prod_{\psi_S} |c_i|$, which represents the worst case number of disjuncts in $\mathrm{DNF}(\psi_S)$. As experimentally demonstrated in that paper, the algorithm can efficiently handle (small) updates of large databases.

[2] The algorithm given there converts $patch(\mu_j)$ to DNF, which is unnecessary, as it is easily seen from theorem 1. Without this conversion, the cost is $O((\prod_{\psi_S} |c_i|)(|\mu| lit_\mu))$, where lit_μ is the total number of literals in μ.

Since the theorem also holds for $\circ_F$, it is easy to see that an algorithm for $\circ_F$ can be designed that, under the assumption that the size of μ is bounded, has the same worst case complexity.

5 Dalal's cardinality based revision

As Forbus' operator $\circ_F$, Dalal's [1988a] revision operator $\circ_D$ uses an ordering induced by the number of propositional letters in which two interpretations differ. But whereas $\circ_F$ collects some models of μ for *each* model of ψ, $\circ_D$ ignores some models of ψ which intuitively are "too different" from the models of μ. Formally, instead of associating an ordering to each model, $\circ_D$ associates a total preorder $\leq_\psi$ to each formula ψ, such that $I \leq_\psi J$ iff $\min_{M \in Mod(\psi)} Dist(M,I) \leq \min_{M \in Mod(\psi)} Dist(M,J)$. The operator $\circ_D$ is then defined by:

$$Mod(\psi \circ_D \mu) = Min(Mod(\mu), \leq_\psi).$$

Example 2 Let $\psi = (a \wedge b \wedge c) \vee (a \wedge \neg b \wedge \neg c)$, and $\mu = \neg a \wedge c$. The models of ψ are $M_1 = \{a, b, c\}$ and $M_2 = \{a, \neg b, \neg c\}$; the models of μ are $N_1 = \{\neg a, b, c\}$ and $N_2 = \{\neg a, \neg b, c\}$. Then $Mod(\psi \circ_D \mu) = \{N_1\}$, since N_1 differs from M_1 in exactly one literal. M_2 does not "generate" any model of the revised database: though $Dist(N_2,M_2) < Dist(N_1,M_2)$, N_2 is ruled out, because $Dist(N_1,M_1) < Dist(N_2,M_2) = Dist(N_2,M_1)$. In contrast, $Mod(\psi \circ_F \mu) = \{N_1, N_2\}$. □

In order to obtain a syntactic characterization of $\circ_D$, we need the following definitions:

$$\begin{aligned}
MinDist(\psi_i, \mu) &= \min_{\mu_j \in \mu} Dist(\psi_i,\mu_j) \\
MinDiff(\psi_i, \mu) &= \{\mu_j \in \mu \mid Dist(\psi_i,\mu_j) = MinDist(\psi_i, \mu)\} \\
DNF_{min}(\psi, \mu) &= \{\psi_i \in \mathrm{DNF}(\psi) \mid \forall \psi_j \in \mathrm{DNF}(\psi) : \\
&\qquad MinDist(\psi_i, \mu) \leq MinDist(\psi_j, \mu)\}
\end{aligned}$$

$MinDist(\psi_i, \mu)$ provides a measure of 'distance' between a conjunction of literals ψ_i and a DNF formula μ; $MinDiff(\psi_i, \mu)$ collects all the $\mu_j \in \mu$ whose distance from a given ψ_i is minimal; finally, $DNF_{min}(\psi, \mu)$ collects the formulas in $\mathrm{DNF}(\psi)$ which fare best in terms of their distance to μ. The following theorem provides a syntactic characterization of Dalal's revision operator.

Theorem 4

$$\psi \circ_D \mu \equiv \bigvee_{\substack{\mu_j \in MinDiff(\psi_i, \mu) \\ \psi_i \in DNF_{min}(\psi,\mu)}} \bigwedge((\psi_i - Diff(\psi_i,\mu_j)) \cup \mu_j).$$

Again, very efficient algorithms can be designed to compute $\circ_D$ for DNF databases or under the model checking approach. Unfortunately, theorem 3 does not hold for $\circ_D$.

Example 3 Let $\psi = (a \vee \neg b) \wedge b$, $\mu = \neg a$. Then $\psi \circ_D \mu \equiv (a \wedge b) \circ_D \neg a \equiv \neg a \wedge b$. But $\mathrm{DNF}(\psi_S) = \{a, \neg b\}$, and $\mathrm{DNF}_{min}(\psi_S, \mu) = \{\neg b\}$, so $\psi_U \wedge (\psi_S \circ_D \mu) \equiv b \wedge (\neg b \wedge \neg a) \equiv$ *false*. In contrast, using theorem 3 we obtain $\psi \circ_F \mu \equiv \psi_U \wedge (\psi_S \circ_F \mu) \equiv b \wedge (\neg a \vee (\neg b \wedge \neg a)) \equiv b \wedge \neg a$. □

Intuitively, the problem lies in the 'model pruning' operation involved in revision. In example 2, M_2 is pruned in the sense that models of μ closest to M_2 are ignored, something which is captured in theorem 4 by the fact that $a \wedge \neg b \wedge \neg c \notin \mathrm{DNF}_{min}(\psi, \mu)$. But as example 3 illustrates, using DNF_{min} on ψ_S instead of ψ might select a set of disjunct *all* of which are inconsistent with ψ. When (and only when) this is the case, using the analogue to theorem 3 for $\circ_D$ will result in an incorrectly revised (in fact, inconsistent) database.

If we think of DNF disjuncts as partial models, the problem is that some of the disjuncts in $\mathrm{DNF}(\psi_S)$ do not represent partial models of the database ψ, and thus there is no need to update or revise these disjuncts. But checking whether this is the case is extremely costly, and thus the restriction to ψ_S can be seen as a way of *guessing* partial models. If every model of the database is changed independently, as in update, the "bad guesses" will generate disjuncts which are inconsistent with ψ_U (as $\neg b \wedge \neg a$ in example 3 with $\diamond_F$) and thus do not result in spurious models of the modified database. In revision, in contrast, we need to prune some models of ψ, and as illustrated by example 3, performing the pruning operation of ψ_S instead of ψ might rule out all partial models of ψ.

We can still develop revision procedures for CNF and NNF databases, though at a substantially higher cost than for update. We do so in two steps. We first bound the set of clauses or top level conjuncts that needs to be considered (theorem 5); then we show that this set can be used to filter the disjuncts of $\mathrm{DNF}(\psi_S)$ in order to compute the revised database (theorem 6). It will then be easy to design an algorithm based on these results.

The set ψ_C of clauses or top level conjuncts that needs to be considered is the set of conjuncts 'connected' to μ in the sense that they share propositional symbols with μ or with another clause connected with μ. Formally:

$$\begin{aligned}
\psi_C^0 &= \psi_S = \{c_i \in \psi \mid Prop(c_i) \cap Prop(\mu) \neq \emptyset\} \\
\psi_C^n &= \{c_i \in \psi \mid Prop(c_i) \cap Prop(\psi_C^{n-1}) \neq \emptyset\} \\
\psi_C &= \psi_C^n \text{ for any } n \text{ such that } \psi_C^n = \psi_C^{n+1}.
\end{aligned}$$

Theorem 5 *Let ψ be an NNF or CNF database. Let ψ_C be as defined above and let $\psi_N = \psi - \psi_C$. Then $\psi \circ_D \mu \equiv (\psi_C \circ_D \mu) \wedge \psi_N$.*

We could thus compute the revised database by using theorem 4 to compute expression $\psi_C \circ_D \mu$ in theorem 5. But computing $\mathrm{DNF}(\psi_C)$, as this procedure would require, is unnecessary. Instead, we can use ψ_C as a filter on ψ_S, in order to remove those disjuncts $\psi_i \in \mathrm{DNF}(\psi_S)$ which, though perhaps at a minimal distance of μ, are not partial models of ψ. This idea can be formalized by making the definition of $\mathrm{DNF}_{min}(\psi_S, \mu)$ depend on ψ_C. Let $\mathrm{DNF}_{min}(\psi_S, \mu, \psi_C)$ be the set of $\psi_i \in \mathrm{DNF}(\psi_S)$ such that:

1. $\exists \psi_k \in \mathrm{DNF}(\psi_S): \psi_C \not\vdash \neg\psi_k$ and $MinDist(\psi_i, \mu) = MinDist(\psi_k, \mu)$; and
2. $\forall \psi_j \in \mathrm{DNF}(\psi_S)$: if $\psi_C \not\vdash \neg\psi_j$ then $MinDist(\psi_i, \mu) \leq MinDist(\psi_j, \mu)$.

Theorem 6 *Let ψ be an NNF or CNF database, ψ_S and ψ_C be as defined above, and $\psi_U = \psi - \psi_S$. Then*

$$\psi \circ_D \mu \equiv \psi_U \wedge \bigvee_{\substack{\mu_j \in MinDiff(\psi_i, \mu) \\ \psi_i \in DNF_{min}(\psi_S, \mu, \psi_C)}} \bigwedge ((\psi_i - Diff(\psi_i, \mu_j)) \cup \mu_j).$$

The following is a simple algorithm to compute $\circ_D$:

Procedure Dalal-Revise(ψ, μ)

1. Make an array Distances of size $|\mu_{max}| + 1$.
2. For each $\psi_i \in \mathrm{DNF}(\psi_S)$, compute $MinDiff(\psi_i, \mu)$ and $k = MinDist(\psi_i, \mu)$, and store ψ_i and $MinDiff(\psi_i, \mu)$ in Distances[k].
3. Traverse Distances in ascending order, until an index m and a disjunct $\psi_i \in$ Distances[m] are found such that $\bigwedge(\psi_C - \psi_S) \wedge \psi_i$ is satisfiable (in which case $\mathrm{DNF}_{min}(\psi_S, \mu, \psi_C) =$ Distances[m]).
4. Return $\psi_U \wedge \bigvee_{\substack{\psi_i \in \text{Distances[m]} \\ \mu_j \in MinDiff(\psi_i, \mu)}} \bigwedge((\psi_i - Diff(\psi_i, \mu_j)) \cup \mu_j)$.

The algorithm is clearly quite similar to the one presented in [del Val, 1992b] for PMA update, so many of the comments made there about efficiency and optimization apply here as well. Under the same implementation assumptions as were made there, it can be shown that the worst case complexity of the procedure, for CNF input and NNF output and ignoring retrieval costs, is $O((\prod_{\psi_S} |c_i|)|\mu| lit_\mu \mathrm{SAT}_{\psi_C})$, where SAT_{ψ_C} is the cost, for any given $\psi_i \in \mathrm{DNF}(\psi_S)$, of testing whether $(\psi_C - \psi_S) \vdash \neg\psi_i$.

As we can see, the crucial difference between this result and the cost of update lies in that there is no need for consistency checks in the latter, a difference which can be directly traced, as mentioned above, to the model-selection operation represented by DNF_{min}. Using $O(\prod_{\psi_C} |c_i|)$ as an upper bound for $O((\prod_{\psi_S} |c_i|)\mathrm{SAT}_{\psi_C})$, an assuming the size of μ is bounded by a constant, the *additional* cost of Dalal's revision with respect to the cost of PMA update is in the worst case $O(\prod_{\psi_C - \psi_S} |c_i|)$, or in essence asymptotically exponential on the size of the difference $\psi_C - \psi_S$. This added cost can be greatly reduced if the database is Horn. In this case, SAT_{ψ_C} will be linear on the size of $\psi_C - \psi_S$ [Dowling and Gallier, 1984; Gallo and Urbani, 1989]. Notice also that in *Dalal-Revise* a huge number of disjuncts might be pruned, resulting in a potentially much smaller database growth than with update operators.

6 Satoh's set-inclusion based revision

Satoh [1988] proposed what can be seen as the natural set inclusion based alternative for revision to Dalal's cardinality-based approach. Define the set $ModMinDiff(\psi, \mu)$ of minimal differences between models of ψ and μ as

$$Min(\{Diff(I, M) \mid M \in Mod(\psi), I \in Mod(\mu)\}, \subseteq).$$

Specialized to propositional logic, Satoh's revision operator $\circ_S$ can now be defined as:

$$\begin{aligned}
Mod(\psi \circ_S \mu) \;\; = \;\; & \{I \in Mod(\mu) \mid \exists M \in Mod(\psi): \\
& Diff(I, M) \in ModMinDiff(\psi, \mu)\}.
\end{aligned}$$

Example 4 Let ψ and μ be as in example 2. Then $Mod(\psi \circ_S \mu) = Mod(\psi \circ_D \mu) = \{N_1\}$. Again, N_2 is closest to M_2, but is not a model of the revised database, since $Diff(N_1, M_1)$ is a strict subset of both $Diff(N_2, M_2)$ and $Diff(N_2, M_1)$. □

The operator $\circ_S$ can be syntactically characterized as follows. A natural syntactic counterpart to $ModMinDiff$ can be obtained by defining $SynMinDiff(\psi, \mu)$ as:

$$Min(\{Prop(Diff(\psi_i, \mu_k)) \mid \psi_i \in \mathrm{DNF}(\psi), \mu_k \in \mu\}, \subseteq).$$

We can then syntactically mimic the model selection operation in the definition of $\circ_S$ by defining:

$$MinPairs(\psi, \mu) = \{< \psi_i, \mu_j > \mid \psi_i \in \mathrm{DNF}(\psi), \mu_j \in \mu, \\ Prop(Diff(\psi_i, \mu_j))) \in SynMinDiff(\psi, \mu)\}$$

Theorem 7

$$\psi \circ_S \mu \equiv \bigvee_{<\psi_i, \mu_j> \in MinPairs(\psi, \mu)} \bigwedge((\psi_i - Diff(\psi_i, \mu_j)) \cup \mu_j).$$

In the case of CNF and NNF databases, it is again not possible to compute the revised database by considering only ψ_S, as can again be shown with example 3. We can solve the problem in a similar way as before. It would suffice to use ψ_C as in theorem 5, but an even tighter result can be obtained by defining a version of $MinPairs$ relative to ψ_C. The idea, very much in the spirit of the definition of $\mathrm{DNF}_{min}(\psi_S, \mu, \psi_C)$, is again to filter out disjuncts $\psi_i \in \mathrm{DNF}(\psi_S)$ which could potentially be taken as elements of a minimal pair, but which are not really partial models of ψ. Formally, define $MinPairs(\psi_S, \mu, \psi_C)$ to be the set of pairs $< \psi_i, \mu_j >$ such that:

1. $\psi_i \in \mathrm{DNF}(\psi_S)$ and $\mu_j \in \mu$;
2. $\exists \psi_k \in \mathrm{DNF}(\psi_S), \mu_l \in \mu : \psi_C \not\vdash \neg\psi_k$, and $Prop(Diff(\psi_k, \mu_l)) = Prop(Diff(\psi_i, \mu_j))$;
3. $\forall \psi_m \in \mathrm{DNF}(\psi_S), \mu_n \in \mu$: if $Prop(Diff(\psi_m, \mu_n)) \subset Prop(Diff(\psi_i, \mu_j))$ then $\psi_C \vdash \neg\psi_m$.

Theorem 8 *Let ψ be an NNF or CNF database, and let ψ_C, ψ_S, ψ_U be as above. Then*

$$\psi \circ_S \mu \equiv \psi_U \wedge \bigvee_{<\psi_i, \mu_j> \in MinPairs(\psi_S, \mu, \psi_C)} \bigwedge((\psi_i - Diff(\psi_i, \mu_j)) \cup \mu_j)$$

It is now easy to obtain a procedure to compute Satoh's revision similar to the one presented for Dalal's revision. As before, we have to compute $\mathrm{DNF}(\psi_S)$. In this case, however, we cannot have the various $\psi_i \in \mathrm{DNF}(\psi_S)$ totally ordered by their distance to μ. We can use instead some data structure which supports efficient subsumption checks to store the various different values of $Prop(Diff(\psi_i, \mu_j))$, storing with each such value the set of all formulas $\bigwedge(\psi_k - Diff(\psi_k, \mu_l)) \cup \mu_l$ such that $Prop(Diff(\psi_k, \mu_l)) = Prop(Diff(\psi_i, \mu_j))$. (For example, we can use a trie as suggested in [de Kleer, 1992], with each terminal node storing all the associated formulas.)

Procedure Satoh-Revise(ψ, μ)

1. $\mathcal{D} := \emptyset$.
2. For each $\psi_i \in \mathrm{DNF}(\psi_S)$ and each $\mu_j \in \mu$ do:
3. $\quad D_{ij} := Prop(Diff(\psi_i, \mu_j))$
4. $\quad$ If there exists $D \in \mathcal{D}$ such that $D = D_{ij}$
5. $\quad$ then store $\bigwedge((\psi_i - Diff(\psi_i, \mu_j)) \cup \mu_j)$ with D
6. $\quad$ else if [$D \in \mathcal{D}$ implies $D \not\subset D_{ij}$], and $\psi_C - \psi_S \not\vdash \neg\psi_i$,
7. $\quad\quad$ then store $\bigwedge((\psi_i - Diff(\psi_i, \mu_j)) \cup \mu_j)$ with D_{ij}
8. $\quad\quad\quad \mathcal{D} := (\mathcal{D} - \{D \in \mathcal{D} \mid D_{ij} \subset D\}) \cup \{D_{ij}\}$
9. Return ψ_U conjoined with the disjunction of all formulas stored with some $D \in \mathcal{D}$.

It is easy to see that under the assumption that the size of μ is bounded by a constant, this procedure has the same worst case complexity as the procedure *Dalal-Revise*, since the number as well as the cost of the subset tests made for each ψ_i will be bounded by a constant as well, and since in both procedures we might need to check every generated disjunct for satisfiability. Needless to say, we expect *Dalal-Revise* to perform much better in practice and to require much fewer satisfiability checks. Notice however the important fact that both procedures benefit equally from the restriction to Horn databases.

7 Other revision operators

Borgida [1985] proposes a revision operator $\circ_B$ such that $\psi \circ_B \mu$ is defined as $\psi \wedge \mu$ if $\psi \wedge \mu$ is satisfiable, or as $\psi \diamond_W \mu$ otherwise, where $\diamond_W$ is Winslett's update operator. In view of theorem 1, it is trivial to obtain a syntactic characterization of $\circ_B$, and we omit it.

Weber [1986] proposes a revision procedure that eliminates first from ψ all symbols in $ModMinDiff(\psi, \mu)$, and then conjoins μ with the resulting database. Let $\Omega = \bigcup ModMinDiff(\psi, \mu)$, and for any model M, let $M - \Omega$ be the restriction of M to letters not in Ω. Weber's revision operator $\circ_W$ is defined by:

$$Mod(\psi \circ_W \mu) = \{I \in Mod(\mu) \mid \exists M \in Mod(\psi) : I - \Omega = M - \Omega$$

Weber provides an alternative characterization of $\circ_W$. Let ψ^p_q stand for the formula obtained by replacing every occurrence of p in ψ by q. For any propositional letter p, let $res_p(\psi) = \psi^p_{true} \vee \psi^p_{false}$; for a set of letters $P = \{p_1, \ldots, p_n\}$, let $res_P(\psi) = res_{p_1}(res_{p_2}(\ldots(res_{p_n}(\psi))))$. Then $\psi \circ_W \mu \equiv res_\Omega(\psi) \wedge \mu$ [Weber, 1986, theorem 5.4]. Weber however provides no method to compute Ω other than by examining all models of ψ and all models of μ, so this falls short of a purely syntactic characterization of the operator. Such characterization is however very easy to obtain in view of the results in the previous section.

Theorem 9 $\psi \circ_W \mu \equiv res_\Omega(\psi) \wedge \mu$, *where* $\Omega = \bigcup SynMinDiff(\psi, \mu)$.

Since the final value of $\mathcal{D}$ in the procedure *Satoh-Revise* is precisely $SynMinDiff(\psi, \mu)$, it is now easy to design an algorithm for $\circ_W$.

8 Discussion

We have provided syntactic characterizations and algorithms for a number of propositional belief change operators proposed in the literature. To our knowledge, in fact, the operators we have characterized include all those proposed in the AI literature not based on conditionals and whose result is independent of the syntactic form of the database. It is easy to see that there are close similarities among all the characterizations, which suggests that these techniques are very general. With the exception of $\circ_W$, belief change can be computed by computing the formula $\delta_{ij} = \bigwedge((\psi_i - Diff(\psi_i, \mu_j)) \cup \mu_j)$ for some or all the $\psi_i \in \mathrm{DNF}(\psi)$ ($\mathrm{DNF}(\psi_S)$ for CNF and NNF databases) and some or all the $\mu_j \in \mu$. For $\circ_D$ and $\circ_S$, but not for $\diamond_W$, $\diamond_F$, and $\circ_B$, some of the ψ_i's have to be pruned, with the resulting impact on complexity for CNF and NNF databases; for $\diamond_W$, $\diamond_F$, and

$\circ_B$, but not for $\circ_D$ or $\circ_S$, δ_{ij} has in some cases to be "patched up" by negating, if possible, some of the literals in other μ_k's in μ. As mentioned in the introduction, this generality suggests that the interest of these characterizations goes beyond the usefulness of the algorithms derived from them.

Related work in update algorithms (such as [Chou and Winslett, 1991; Forbus, 1989; Grahne and Mendelzon, 1991]), all of which, unlike ours, assumes that the models of the database are directly available, is discussed at some length in [del Val, 1992b]. As for revision operators, no strictly syntactic characterization of Weber's and Borgida's approach for arbitrary databases and update formulas was previously known. Satoh [1988] provides a second order characterization of $\circ_S$ as defined for predicate calculus, but no same-order characterization was known. Dalal [1988a; 1988b] characterizes his operator $\circ_D$ as follows. Let $G(\psi) = \bigvee_{p \in Prop(\psi)} res_p(\psi)$, $G^0(\psi) = G(\psi)$, $G^n(\psi) = G(G^{n-1}(\psi))$. Dalal shows that $\psi \circ_D \mu \equiv G^k(\psi) \wedge \mu$ for the least k such that the right hand side is satisfiable. Thus, we can compute $\psi \circ_D \mu$ by repeatedly applying the function G to the database resulting from the previous step, checking at each step the consistency of the result with μ. Our method also requires (different) satisfiability checks; but since the function G does not preserve the property of being Horn Dalal's method, unlike ours, cannot take advantage of a restriction to Horn databases. We also note that [Dalal, 1988b, theorem 5.3] provides a method which computes the revised database *given* the minimal distance in which models of μ and ψ differ. The value m used in step 1 of *Dalal-Revise* as an index into the array Distances is just such minimal distance, a fact that we can use in order to apply this last method.

An important area for further work is belief change in the presence of constraints or "protected formulas." We have characterizations of the result of applying each of the operators discussed in the presence of protected formulas, which are identical or very similar to those reported in [del Val, 1992b] for PMA update, which again suggests that similar techniques can be shared accross belief change operators. Another area for further work is the extension of our results to predicate calculus, for which we conjecture that some of the techniques presented in [Chou and Winslett, 1992] can be adapted to deal with partial models (DNF disjuncts) and incorporated into the algorithms of this paper.

References

[Alchourrón *et al.*, 1985] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet functions for contraction and revision. *Journal of Symbolic Logic*, 50:510–530, 1985.

[Borgida, 1985] Alex Borgida. Language features for flexible handling of exceptions in information systems. *ACM Transactions on Database Systems*, 10:565–603, 1985.

[Chou and Winslett, 1991] Timothy SC Chou and Marianne Winslett. Immortal: A model-based belief revision system. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, 1991.

[Chou and Winslett, 1992] Timothy SC Chou and Marianne Winslett. A model-based belief revision system. Submitted, 1992.

[Dalal, 1988a] Mukesh Dalal. Investigations into a theory of knowledge base revision: Preliminary report. In *Proceedings of the Seventh Conference of the AAAI*, 1988.

[Dalal, 1988b] Mukesh Dalal. Updates in propositional databases. Technical Report DCS-TR-222, Computer Science Department, Rutgers University, February 1988.

[de Kleer, 1992] Johan de Kleer. An improved incremental algorithm for generating prime implicates. In *Proceedings of the Tenth Conference of the AAAI*, 1992.

[del Val, 1992a] Alvaro del Val. Approximate belief update. In *Proceedings of the First IEEE Workshop on Imprecise and Approximate Computation*, 1992.

[del Val, 1992b] Alvaro del Val. Computing knowledge base updates. In *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*, 1992.

[del Val, 1993] Alvaro del Val. *Belief Revision and Update*. PhD thesis, Stanford University, 1993.

[Dowling and Gallier, 1984] William F. Dowling and Jean H. Gallier. Linear-time algorithms for testing the satisfiability of propositional horn formulae. *Journal of Logic Programming*, 3:267–284, 1984.

[Forbus, 1989] Kenneth D. Forbus. Introducing actions into qualitative simulations. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, 1989.

[Gallo and Urbani, 1989] Giorgio Gallo and Giampaolo Urbani. Algorithms for testing the satisfiability of propositional formulae. *Journal of Logic Programming*, 7:45–61, 1989.

[Gärdenfors, 1988] Peter Gärdenfors. *Knowledge in Flux*. The MIT Press, 1988.

[Grahne and Mendelzon, 1991] Gösta Grahne and Alberto O. Mendelzon. Updates and subjunctive queries. Technical Report KRR-TR-91-4, Computer Science Department, University of Toronto, July 1991.

[Halpern and Vardi, 1991] Joseph Halpern and Moshe Vardi. Model checking vs. theorem proving: A manifesto. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, 1991.

[Katsuno and Mendelzon, 1991a] Hirofumi Katsuno and Alberto O. Mendelzon. On the difference between updating a knowledge database and revising it. In *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning*, 1991.

[Katsuno and Mendelzon, 1991b] Hirofumi Katsuno and Alberto O. Mendelzon. Propositional knowledge base revision and minimal change. *Artificial Intelligence*, 52:263–294, 1991.

[Satoh, 1988] Ken Satoh. Nonmonotonic reasoning by minimal belief revision. In *Proceedings of the International Conference on Fifth Generation Computer Systems*, 1988.

[Weber, 1986] Andreas Weber. Updating propositional formulas. In *Proceedings of the First International Conference on Expert Database Systems*, 1986.

[Winslett, 1988] Marianne Winslett. Reasoning about action using a possible models approach. In *Proceedings of the Seventh Conference of the AAAI*, 1988.